BAB II
KAJIAN PUSTAKA

2.1 Kontribusi Penelitian

Perbedaan yang paling mendasar antara penelitian sebelumnya dengan penelitian yang akan dilakukan yaitu terdapat pada basis pengetahuan yang dipakai. Dalam tabel 2.1 bisa dijelaskan sebagai berikut.

Tabel 2. 1 Kontribusi penelitian yang dilakukan

Nama penulis	Judul Penelitian	Decision Tree	Realtime	Fail2ban	DDOS	Brute Force
Rudi Rinaldi, Agus Urip Wibowo, dan Yuli Fitrisia	Analisa kinerja *fail2ban* dan *denyhosts* dalam mengamankan *server* dari serangan *brute force*			√		√
Zhuroto, John Friadi	Membangun sistem keamanan komputer untuk menghadapi serangan *brute force* dengan menggunakan *fail2ban*			√		√
Iwan Kurniawa	Sistem pencegahan			√		√

n, Ferry Mulyanto, dan Fuad Nandiasa	serangan *brute force* pada ubuntu *server* dengan menggunakan *fail2ban*					
Eko Ari Irawan	Klasifikasi log serangan Bruteforce dan DDOS Pada Fail2ban Menggunakan Algoritma Decision Tree	√	√	√	√	√

2.2 Teori Penelitian

2.2.1 Serangan Bruteforce

Serangan *brute force* adalah algoritma mencoba memecahkan masalah dengan sederhana, langsung, dan cara itupun bisa dilakukan dengan jelas. Dengan mencoba menyelesaian atau mencari *password cracking* yang valid, serangan *brute force* akan menempatkan atau mencari semua kemungkinan *password* yang sudah disediakan dengan masukan karakter dan panjang *password* tertentu hal ini mencoaba untuk mengkombinasi *password* [2].

2.2.2 Serangan DDOS

Sebuah serangan yang menggunakan *DoS* namun terdistribusi sehingga serangan ini melibatkan banyak komputer yang saling terhubung dan melakukan serangan pada target komputer, *host* yang melakukan serangan *DDoS* disebut

komputer *Zombie*. Dengan cara dilakukan remote dari sebuah *master* yang natinya master akan memerintahkan kepada kumpulan *zombie* komputer untuk menyerang target yang perintahkan. Dengan kata lain serangan ini dilakukan dengan cara beramai ramai yang menyebabkan banjirnya *request* dari *user* dan server tidak mampu melayani *user*[3].

Serangan *DDOS* akan menyerang sebagian *resource* komputer yang dipakai melakukan *reply* untuk layanan *request* biasanya di proses oleh *CPU*. Dengan demikian target tidak akan bisa memberikan ketersediaan *CPU, Memory,* Ataupun penyimpanan sebuah sistem [10]. *DDoS* memakan banyak proses sistem karena menggunakan banayak agen atau host komputer yang dipakai menghabiskan resource sitem tersebut. Ketika serangan berlangsung sistem komputer akan menjadi sangat sibuk karena kebanjiran request. Bonet di gunakan untuk melancarkan serangan *DDoS Flooding* dalam proses *attacking* serangan ini terdapat beberapa peranan seperti Master, Handler, Agen dan korban.

Dalam sebuah contoh terdapat sebuah *website* yang sering di kunjungi ketika di lakukan DDoS maka website itu akan melemah, memberikan informasi sangat lambat sampai tidak bisa memberikan informasi pada *client*.

2.2.3 Fail2ban

Fail2ban merupakan paket program untuk mendeteksi usaha login yang gagal dan kemudian memblokir alamat *IP host* asal (Fail2ban.org, 2014). *Fail2ban* melakukan *scanning* terhadap file log (misalnya /var/log /apache/error_log) dan melarang IP yang menunjukkan tanda-tanda berbahaya . Kondisi bahaya yang dimaksud adalah seperti terlalu banyak kegagalan *password*, mencari eksploitasi, dan lain-lain. Secara umum *fail2ban* kemudian digunakan untuk memperbarui aturan-aturan firewall untuk drop (menolak) alamat IP tertentu dalam kurun waktu tertentu. Disamping itu *fail2ban* dapat dikonfigurasi untuk mengirim email notifikasi jika ada serangan yang berhasil digagalkan. *Fail2ban* dilengkapi dengan filter untuk berbagai layanan, seperti *apache*, *mail*, *ssh*, dan lain-lain [11].

Fail2ban dirancang untuk membuka atau meberi hak akses host yang diblokir dalam jangka waktu tertentu, sehingga fail2ban ini tidak mengunci setiap koneksi yang mungkin sudah terkonfigurasi sementara. Namun, menghentika ip address

dalam waktu tertentu, *fail2ban* digunakan untuk menghentikan koneksi jaringan yang sedang membanjiri sistem jaringan, serta mengurangi kemungkinan suksesnya untuk melakukan dictionary attack.

2.2.4 Fail2sql

Fail2sql digunakan untuk mencatat informasi ke *database MySQL* termasuk lokasi *geografis*. Informasi ini kemudian dapat digunakan dalam laporan, grafik atau oleh program pihak ketiga untuk mengambil tindakan lebih lanjut seperti pemblokiran permanen, pelaporan ke ISP dan sebagainya [12].

2.2.5 SSH

Secure Shell (*ssh*) adalah aplikasi untuk mengakses jarak jauh aman. Dikembangkan pertama kali dikemukanan OpenBSD project dan selanjutnya munculah versi rilis (port) dimanage oleh team porting ke sistem operasi lainnya, termasuk sistem operasi *Linux*Fungsi ssh untuk mengakses mesin jarak jauh secara *remote*. Scp yang merupakan bagian dari *SSH* yang berfungsi untuk mengganti rcp yang lebih aman, keluarga lainnya adalah *sftp* yang dapat digunakan sebagai pengganti *FTP* [13]. Menggunakan *ssh*, maka semua percakapan antara *Server* dan *client* di-enkripsi lebih aman karena jika percaban disadap tidak mengerti konteksnya. Seandainya seorang administrator sedang melakukan perbaikan *Server* dari jauh dengan menggunakan *ssh*, tentunya dengan *account* yang memiliki hak yang sama pada admin, dan tidak sepengetauan admin, account dan *password* tersebut disadap oleh pihak luar, kemudian *server* diretas [13].

2.2.6 IPTables

IPTables merupakan *firewall* yang tersedia di sistem operasi *Linux* dan mudah untuk digunakan. Dasar dari *IPTables* hanya mendasari konsep *TCP*/*IP*. *IPTables* terdiri dari tiga macam daftar aturan yang sudah baawaan dari sistem operasi *Linux*.

2.2.7 HTML

HTML singkatan dari *Hypertext Markup Language* yang merupakan suatu bahasa dari *World Wide Web*. Format *HTML* dapat ditemui saat mengakses internet atau homepage. Semua format hyperlink yang dapat diklik seperti

gambar, grafis, dokumen multimedia, form yang dapat diisi, dan segala sesuatu yang terdapat pada homepage dibuat berdasarkan perintah atau *syntax* HTML. Terdapat beberapa contoh tag HTML sebagai berikut : [14].

<HTML> dan </HTML>

<HEAD> dan </HEAD>

<TITLE> dan </TITLE>

<BODY> dan </BODY>

<P>, dan lain sebagainya.

2.2.8 PHP

Bahasa pemrograman PHP adalah *script* untuk pemograman yang ada pada *web server side* yang membuat dokumen HTML secara *on the fly*.Dengan menerapkan bahasa pemrograman PHP maka maintenance suatu situs dari *web* menjadi lebih mudah seperti penggunaan proses *update* data. PHP dikenal pertama kali dengan sebutan PHP/FI yaitu PHP – Personal *Home Page* FI (*Form Interface*). Diciptakan oleh Rasmus Lerdoff. PHP awalnya adalah merupakan program CGI untuk menerima input dari form yang terdapat pada suatu *web browser* [15].

Software ini dipublis dan dilisensikan sebagai perangkat lunak *OpenSource*.Secara resmi, PHP merupakan kependekan dari *PHP:HyperText Preprocessor* [15]. Berikut adalah contoh potongan *script* PHP:

```
<html>
<head>
<title> Serangan </title>
</head>
<body>
<?php
    Echo "Data Serangan";
?>
</body>
</html>
```

2.2.9 JavaScript

Pada tahun 1995, *JavaScript* awalnya diperkenalkan Netscape. *JavaScript* adalah bahasa yang dapat berjalan pada *web browser* sebagai *client server programming* serta digunakan untuk menyediakan akses script untuk objek yang dimasukkan (*embedded*) di aplikasi lain. *JavaScript* digunakan pada browser atau navigator yang memanggil halaman *website* berisi *script-script*. *JavaScript* ini tidak membutuhkan kompilator atau penerjemah khusus untuk menjalankannya, karena kompilator yang ada pada *JavaScript* sendiri sudah termasuk di dalam browser itu sendiri [16].

JavaScript telah banyak digunakan dalam perkembangan pemrograman *web* pada sisi *client* dewasa ini. Dengan mengimplementasikan *JavaScript*,*web* akan menjadi lebih ringan, cepat, dan tampilan lebih baik dengan menambahkan animasi [17].

2.2.10 Decision Tree

Decision Tree digunakan untuk menentukan jenis serangan terhadap server, decision tree merupakan struktur pohon dan setiap *node* dalam pohon akan merepresentasikan atribut yang telah diuji, dan dalam *decision tree* setiap cabang adalah suatu pembagian hasil uji dari beberapa *node*, *node* daun mewakili kelompok kelas tertentu. Level *node* teratas adalah *node* akar atau biasanya disebut *root* yaitu berupa atribut yang paling memiliki dampak paling besar pada suatu kelas tertentu. *Decision Tree* juga melakukan strategi pencarian secara *top-down* untuk mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari *node* akar atau yang disebut *root* sampai *node* akhir dan kemudian hasilnya akan diprediksi kelas yang mendapatkan data baru tertentu[18].